Performance evolution of Hypercube Computers through Simulation

Y. JAYANTA SINGH
S. C. MEHROTRA

Abstract

A parallel processing system is such a system that a large amount of work is distributed equally among many processors and are made to work for a single solution. Hypercube computers are used for parallel processing. Hypercube computers are introduced under different names like cosmic cube, n-cube, binary n-cube, boolean n-cube etc. The some of commercial hypercube available are binary hypercube, like, iPSC/2, nCUBE, and Mark III etc. The excellent performance of this type of computer on complex applications and its modest cost and open-ended expandability give an effective means of achieving faster and cheap computing in the near future. For example a 64-node machine has five to ten times faster than a VAX1 l/780. Due to their structural regularity and high potential for the parallel execution of various algorithms, hypercube computers have drawn considerable attention in recent years from both Academic and industrial communities.

A simulation study had performed on hypercube computer to bring the optimization for (i) the distribution timing of workloads among the neighbor processors from server or originating processor, (ii) computing time and (iii) effects of different setup timing of processors on total processing time.

Key Words : Parallel processing, Hypercube, computing time, setup timing ,

1. INTRODUCTION

Parallel computers have a very high processing speed. If one computer needs time T to solve a given problem, then P processors should need only time T/P for the same problem. If the cost to introduce a parallel computer with P processor is less than P times the cost of a single processor computer then speed increase and cost reduction can be obtained [Fadi N. Sibai (1998)]. Hypercube computers are uses for parallel computing. Several research and commercial such as Intel, NCUBE, Floating Point System, Ametek, and Thinking Machine have been built hypercube computers. [John P.H and Trevor M(1989), H.Y.Chang and R. J Chen (2001), Charles L.Seitz(1985)].

The structure of Hypercube is a loosely couple parallel processor based on the binary n-cube network. An n-cube parallel processors, contains of 2n identical processors, each provide with its own sizable memory and interconnected with n neighbors. The hypercube architecture has four major characteristics: (i) High degree of connectivity (ii) Less diameter (iii) Simple routing mechanism and (iv) Isomorphic processors.

In this paper, section 2 describes the working principles of Hypercube. Section 3 discuss about Hypercube parameters. A simple simulation model is discussed in section4. Section5 analyses the different new outcomes of the study using simulation.

2. WORKING PRINCIPLE

The basic concept and working principle of a Hypercube can be discus as following. An n-dimensional hypercube graph Qn contains N=2n nodes of degree n, and n*2 n-1 edges. If the numbers of the nodes N is increased to improved performances, the requirements of the node increase at a rate proportional to n=log2N. This shows that practical limitations on the number of links per node can be met. The maximum internodes distance or diameter of Qn is n, which defines the worst-case communication delay [Y. C.Chebg and T.G. Robert (1988)]. The average inter node distances is (n*2 n-1)/(2 n-1).

Programming on hypercube computers are done by writing a separate program to run on each processors. These programs are the copies of a single program and the distinct copies will execute correctly regardless of their location in the hypercube. This style of programming is referred to as single code multiple data or single program multiple data. The selection of path is according to sequence of Gray code. The developments in VLSI and new IC technologies made hypercube computers feasible in the first place to reshape this machine and lead to further improvements in their performance/cost ratio [John P.H and Trevor M (1989)]

Construction

A zero-dimensional hypercube has a single processor; a one-dimensional hypercube is constructed by connecting two zero-dimensional hypercube and so on. In general a (K-1)-dimensional hypercube is constructed by connecting the corresponding processors of two k-dimensional hypercube [Y.S.Chen, 1996]. See figure 1 and figure 2.

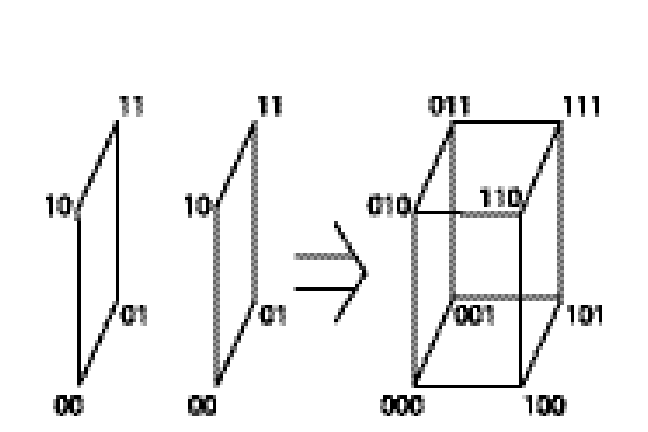


Figure 1. Recursively building the 3 Cube from 2-2 Cubes

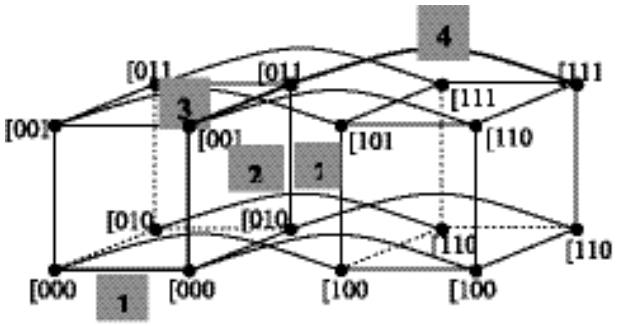


Figure 2 a 4-Dimensional Hypercube

Data Transfer

Let A and B are any two nodes of a Hypercube and consider lets consider data are sending from node A to node B. The data may move along a path from A to B crossing a number of processors. The length of a path between two nodes is the simply the number of edges of the path. The minimum distance between the nodes A and B is equal to the number of the bits that differ between A and B that is the Hamming distances [Y. Saad and M.H. Shultz (1998)]. A processor can be reached from any other processor in several ways. To fix one path only, the easiest way is to consider the originating processor as the root and all other processors are nodes at each level of the spanning tree. When any two paths lead to the same processor, prefer the path, which is connected to lesser number of processors so that the communication can be faster. Always consider the minimal hops between the originating processor and any other processor, and choosing the path having lesser number of processors.

3. HYPERCUBE COMPUTER PARAMETERS

Hypercube uses two classes of different multiple instruction multiple data (MIMD) parallel processors i.e. message passing and direct connection. Interconnection is achieved by message passing. The advantage of this class is the ability to exploit particular topologies of problems in order to minimize communication cost. The second class of parallel processors consists of a set of N identical processors interconnected via a large switching network to N memories. In such system when a processor wants to send some data to a neighboring processor, it incurs a setup time, for setting up of DMA channels and buffering delays, involved in the inter-processor communication of the data. The main advantage of this class is that it makes the data access transparent to the user [Y. Saad and M.H. Shultz (1998), Robert C. Bedichek (1994), Jahng S. Park (1991)]. A sample structure of a hypercube network is given in figure 3.

The important considerable parameters for any interconnected network are degree of a node, diameter, average distance, cost(=degree*diameter), bisection bandwidth, connectivity, fault-tolerance, reconfiguration, symmetry, excursiveness, scalability, and Hamilton path or cycle etc[Guihai Chen (1984)]. Three most important hypercube parameters discuss in this study are given below.

I) DEGREE OF A NODE

Degree or connectivity of a node in a multiprocessor system is defined as the number of connections at every node and determines the hardware complexity of the network. The higher the connectivity, the higher is the hardware complexity and hence the cost of the network.

II) DIAMETER:

The diameter of a multiprocessor system is defined as maximum {dab} where dab is the shortest distance between nodes a and b. The diameter of multiprocessor system is an important parameter in determining the performance of the network [Kemal Efe (1991)]

III) SETUP TIME

In most of the available literature the 'setup time' was neither considered nor included with the communication time properly. When a data is distributed among the processors, it is probable that the time to communicate a fraction of data can be of the order of setup time. This is considerable when a DMA communication is used.

4. A HYPERCUBE COMPUTER SIMULATION MODEL

The primary tool used in this study is a Simulation process. In the simulated environment all things are ultimately knowable easily without disturbing the measured environment but in the real world, most of the measurement tools are very noisy, troublesome and extremely difficult to observe. Prediction and optimization of behavior of complex real world systems are performed with writing computer programs i.e. simulation model. A preliminary simulation on hypercube computer has been performed based on:-

1. distribution timing of workloads among the neighbor processors from an originating processor,

2. computing time and

3. effects of different setup timing of processors on total processing time.

In this model the passing message holds the operation and com-

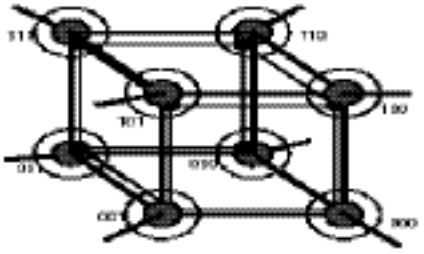
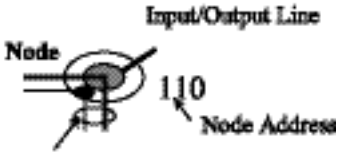


Figure: 3. a 3-D hypercube network



munication between them. The message passing performance is measured in units of time or bandwidth i.e. bytes per second [Stevn D.Y(TM-4380), E.R. Jessup2 (1995), Ion S, F Sultan, David K(1996)]. The time for a small, or zero length, messages is bounded by the speed of the signal through the media and any software in sending or receiving the message. Message passing time is usually a linear function of message size for two processors that are directly connected. For more completed networks, per hop delay may increase the message passing time. The message passing time "Tn" can be models as

$$Tn = a + bn + (h-1)g \qquad \text{.....(i)}$$

Where 'a' is start-up time, 'b' is per byte cost, 'n' is number of bytes per message, 'h' is number of hops a message, must travel, and 'g' is per hop delay [Jack.J.D, Thomas.H.D (1996)]. Deferent versions of the iPSC have different values of above parameters. The study considered the value of the parameters of iPSC/860, which is mention in work of [Thomas H.D (1994)]. In this model, the value of a is 136 ms, b is 0.4 and g is 33. A simple hypercube system is simulated by using GPSS World Simulator [GPSS World Simulator 2001], using the above parameter values. The results of the simulation are discussed below.

5. ANALYSES OF THE NEW RESULTS OF THE SIMULATION STUDY

The performance of the hypercube system are discussing below. These are only the statistical values, not the exact values.

5.1 Distribution timing of workloads

When a processor wants to send some data to neighboring processors, each of the processor of a hypercube system needs setup timings (in m sec.) This timing is required for setting up of DMA channels and buffering delays involved in the interprocess communication of the data. The flows of a slot of workload among processors are according to gray code order [E.R. Jessup1 (1995)]. It is sure that the processors, which are near to the originating processor (server), will take less time to receive its share than the processor those are at far position. This is shown in figure 4.

Other than the computing speed, another important factor is the

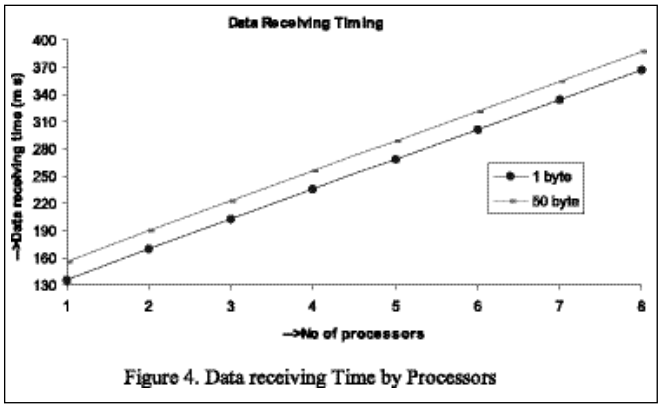


Figure 4. Data receiving Time by Processors

data receiving by the processor to start processing. The study considered two small workloads (say 1 byte and 50 bytes) in order to analyze the data receiving time by the processors (closer or far from originating processor).

Here each of the processor has different data receiving timings depending on its distance from the originating processor. As this distance increased the data receiving time is also become larger. This is because the message or data may be moved along a path crossing a number of processors. And the data receiving time also become larger, as larger the workload. In the example, the movement of 50 bytes takes more time then that of time of 1 byte of data (according to theory). However if the workload is small enough, it does not have meaning to use more processors.

5.2. Computing time

A large workload is distributed among number of processors of a hypercube system. All the processors have computed on a slot of a single job, concurrently. The computing timing is always reduced with increase in the numbers of the processors. How large the supplied workload, performance of the entire system depends on number of processors that are linked to the system. This is illustrated in figure 5

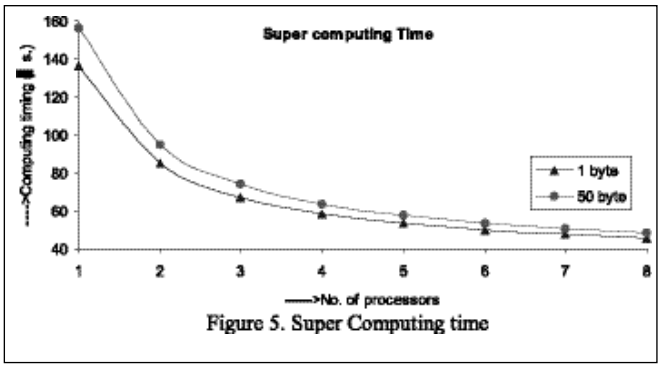


Figure 5. Super Computing time

5.3 Effects of Setup timing of processors

For calculating the performance of the hypercube system, the setup timings of the processors plays a main role. At lowering this setup timing increased the performance and the entire computational timing are reduced. Reducing the setup timing can be done in hardware part by synchronizing the clock timing [E.R. Jessup2(1995), Thomas H.D,(1994)]. A performance report is shown in Figure 6, for different set values of setup time.

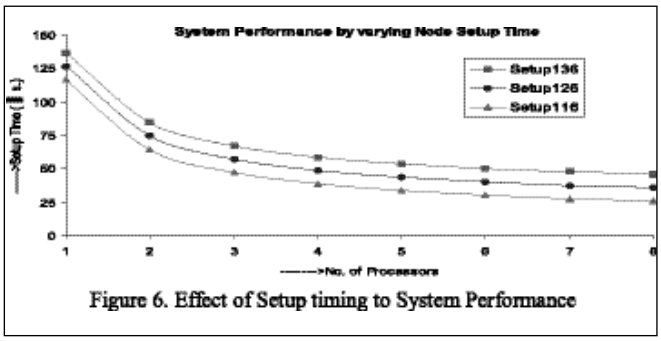


Figure 6. Effect of Setup timing to System Performance

6. CONCLUSION

In designing of hypercube computers and proper arrangement of associated processors, packaging considerations are very important in order to keep its physical size, computation speed within the reasonable limits. This is due to large number of nodes that may be present in a Hypercube.

1. Each processor needs a "setup time" to start any work. If an apply workload is small, there is no need of using high dimensional hypercube. In such a system, the communication timing among the number of processor may take more long time than actual computational time. However, for a large workload, higher number of processor will give a faster processing speed. If the setup time for the processor is reduced to some lower range than more optimum performance can be achieved.

2. Again the performance of the entire system depends on number of processors are linked to the system.

3. In case of a linear network, the total processing time is minimized when the originating processor is at the center of the network. The choice of the network interface can also dominate the system performance. Several of study can be done to analyze or enhance the speed of the processing of hypercube.

The reason of using the simulation is because in the real world, most of the measurement tools are very noisy, troublesome and extremely difficult to observe, but in the simulated environment all things are ultimately knowable easily without disturbing the measured environment. It is mostly very hard to reconfigure real systems. This simulator can easily be reconfigured to reflect the changes of the simulated architecture. Existing registered software and pieces of hardware have a particular maximum performance, which cannot be overcome. It is rather easy in simulation of such architectures to analyze of performance of each parameter.

There may come a very small (negligible) difference in the results of a simulation environment with that of actual environment.

REFFERENCES

Charles L.Seitz (1985). "The Cosmic Cube", Communication of ACM, vol. 28(1)

E.R. Jessup1(1995). "Using the iPSC/2 at CU Boulder, (HPSC), Univ. of Colarado

E.R. Jessup2(1995). "Distributed-Memory MIMD Computing: An Introduction",High Performance Scientific Computing (HPSC), Univ. of Colarado

Fadi N. Sibai (1998). "The Hyper-ring Network: A cost efficient Topology for Scalable Multicomputers", Intel Corporation, ACM

Guihai Chen (1984). "A tutorial on Interconnection Networks", IEEE C. Society GPSS World Simulator(2001). NC,US, Munutesman Software (4E) www.minutesmansoftware.com

H.Y.Chang and R. J Chen(2001). "Graph Embedding Aspect of IEH Graphs", Journal of Inf. Sc. And Engg. Vol. 17, 23-33.

Ion S, F Sultan, David K(1996). " A Hyperbolic Model for Communication in Layered Parallel Processing Environments", J. of Parallel & Distributed Computing.

Jack.J.D, Thomas.H.D(1996). "Message passing Performance of various computers", Oak Ridge National Lab, ORNL/TM-13006.

Jahng S. Park(1991). "The Folded Hypercube ATM Switches", Ph. D Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia.

John P.H , Trevor M(1989)."Hypercube Supercomputers", Proc.of. IEEE Vol.77(12).

Kemal Efe(1991). "A variation on the Hypercube with Lower Diameter", IEEE Transactions on Computers, Vol. 40, (11).

Robert C. Bedichek(1994). "The Meerkat Multicomputer: Tradeoffs in Multicomputer Architecture", Ph. D Thesis, University of Washington.

Stevn D.Y, "Generalized Hypercube Structure and Hyperswitch Communication Network", NASA, Hapton, TM-4380.

Thomas H.D(1994). "Early Experience and Performances of the Intel Paragon", Oak Ridge National Lab, ORNL/TM-12194

Y. C.Chebg and T.G. Robert(1988). " Distributed Computation with Communication Delay", IEEE Trans.on Aerospace and Electronic systems, Vol. 24, (6)

Y.S.Chen(1996), "Algorithm-Based Fault-Tolerant Strategies in Faulty Hypercube & Star Graph Multicomputers", Ph. D Thesis, National Central University, Taiwan.

Y. Saad and M.H. Shultz(1998), "Topological Properties of Hypercube", IEEE Transactions on Computers, Vol. 37 (7).

Dr. Y. JAYANTA SINGH
Lecturer
Department of Information Technology
Skyline College, P.O.Box. 1797
Sharjah, UAE,
jsingh@skylinecollege.info

Dr. S. C. MEHROTRA
Professor
Department of Computer Science and I.T
Dr. B. Ambedkar Marathwada University.
Aurangabad - 431004. (MH). India.
mehrotrasc@rediffmail.com